Multicore and Cloud-Based Solutions for Genomic Variant Analysis

Cristina Y. González¹, Marta Bleda^{1,2}, Francisco Salavert^{1,2}, Rubén Sánchez¹,
Joaquín Dopazo^{1,2,3}, and Ignacio Medina^{1,3}

¹ Computational Genomics Institute,

Centro de Investigación Príncipe Felipe (CIPF), Valencia, Spain

² CIBER de Enfermedades Raras (CIBERER), Valencia, Spain

³ Functional Genomics Node, Instituto Nacional de Bioinformática (INB) at CIPF,
Valencia, Spain

{cgonzalez,mbleda,fsalavert,rsanchez,jdopazo,imedina}@cipf.es

Abstract. Genomic variant analysis is a complex process that allows to find and study genome mutations. For this purpose, analysis and tests from both biological and statistical points of view must be conducted. Biological data for this kind of analysis are typically stored according to the Variant Call Format (VCF), in gigabytes-sized files that cannot be efficiently processed using conventional software.

In this paper, we introduce part of the High Performance Genomics (HPG) project, whose goal is to develop a collection of efficient and open-source software applications for the genomics area. The paper is mainly focused on HPG Variant, a suite that allows to get the effect of mutations and to conduct genomic-wide and family-based analysis, using a multi-tier architecture based on CellBase Database and a RESTful web service API. Two user clients are also provided: an HTML5 web client and a command-line interface, both using a back-end parallelized using OpenMP. Along with HPG Variant, a library for VCF files handling and a collection of utilities for VCF files preprocessing have been developed.

Positive performance results are shown in comparison with other applications such as PLINK, GenABEL, SNPTEST or VCFtools.

Keywords: Multicore, OpenMP, web service, genomic variant analysis, mutation.

1 Introduction

The application of Next Generation Sequencing (NGS) technologies is uncovering an unexpected amount of genomic variability. Detecting causal mutations related with particular disorders has become a big challenge given that most variants in patients are likely to be neutral [1]. Thus, genomic variant analysis requires an exhaustive cleaning of the generated data together with the application of several analysis and statistical tests.

Biological data for this kind of analysis are typically stored in files according to the Variant Call Format (VCF) [2], a plain text format with very concrete

information about each variant, such as: the chromosome and position where it has occurred, the bases in the reference genome and sequenced samples, and diverse information about the quality, coverage and other parameters of the reading. The size of these files varies from some megabytes to several gigabytes, in some cases impossible to be processed using standard workstations. In the near future, their size will probably increase in an order of magnitude, making this situation even more critical.

The goal of the High Performance Genomics project [3] is to develop a collection of efficient and open-source software applications for the genomic-scale data analysis. This paper is focused on HPG Variant, a suite that allows to get the effect of mutations via a RESTful web service architecture, and to conduct genomic-wide and family-based variant analysis. It avoids some limitations present in other available tools in relation to: (i) time, taking advantage of the parallel capabilities of current hardware in order to improve performance and (ii) memory usage, managing memory carefully in order to allow to process files of arbitrary size.

Along with the HPG Variant suite, a library for VCF files handling was developed. It provides an API also used from HPG VCF Tools, a collection of utilities for VCF file preprocessing. Both the library and the utilities are briefly described.

In order to check the suitability of these applications, some benchmarks have been conducted in comparison to others such as VCFtools [2], PLINK [4,5], GenABEL [6] and SNPTEST [7].

2 HPG Variant

HPG Variant is a suite of tools for performing tests and analysis of genomic variants. It currently consists of two main parts, a variant effect prediction tool named *effect*, and another one oriented to genome-wide association analysis (GWAS) named *gwas*.

Effect is, as its name points, a tool for predicting the effect or consequence type of variants. The effect of a genomic variant measures the putative biological consequence of a mutation, i.e. mutations can be located outside or inside genes or functional DNA regions, thus having different biological consequences. HPG Variant *effect* ports the existing VARIANT tool [8] to a more efficient implementation, and consists of a system with multiple layers, the lowest being CellBase DB [9], which stores information about variants and their consequence types for many species. This database can be queried through a complete set of RESTful web services using multiple user clients.

GWAS includes parallel implementations of several genomic analysis. During the past years, the application of these analysis in DNA microarray-based studies has been determinant in the identification of causal variants [10]. Unfortunately, the new sequencing technology is producing such amount of data that it cannot be easily analyzed using traditional programs. HPG Variant *gwas* tool takes as

a reference some tests available in PLINK, improving their performance both in execution time and memory usage, as well as using standard file formats in the Next Generation Sequencing context. The implemented features include genomic-wide association analysis such as chi-square and Fisher’s exact test, and family-based analysis such as transmission disequilibrium test (TDT). Mendelian errors are automatically removed during the analysis.

The following sections introduce both the architecture and the user interface of HPG Variant.

2.1 Architecture

HPG Variant consists of a 3-layered architecture with a C back-end, a RESTful web services API and a biological database known as CellBase. The architecture design of the last two has been specially focused on fault-tolerancy, thus providing a high-availability solution with no single point of failure. The architecture is shown from a global perspective in Fig. 1.

Fig. 1. HPG Variant *effect* architecture

CellBase, hosted and maintained at Príncipe Felipe Research Center database servers, stores more than 200 GB of data about multiple species, extracted from several biological databases and sources. Database optimizations allow not to spend more than 5 milliseconds on any query. In order to provide a high availability and load balancing, it has been implemented using a MySQL replication cluster, with Keepalived¹ and HAProxy² services configured. For the sake of simplicity, in Fig. 1 it has been represented as one pair server-database.

CellBase is not remotely accessed via direct queries but through specific RESTful API calls [11] that composes those queries. This provides a homogeneous yet still efficient way of retrieving information. The web services layer has been implemented in Java using the Java Jersey library³ and the database is locally accessed using Hibernate⁴.

The back-end is implemented according to both task and data parallelization schemes. Using OpenMP sections, it executes 3 main threads in a double producer-consumer. The application flow is shown in Fig. 2, where each “alt” section corresponds to one of these threads. These “alt” sections apply to all the HPG Variant tools, but the inner “worker threads” section is specific to HPG Variant *effect*.

The first thread is responsible of reading the VCF file. One of the main problems faced during the implementation of the back-end was that VCF files can be several gigabytes-sized and impossible to manage in a single iteration. For this reason, reading was implemented in several steps, each one retrieving a certain amount of lines from the file and queueing them into a list shared between threads. This list controls memory usage by pausing insertions when the maximum authorized size is reached, and resuming when at least one position is free. Optionally, the thread here described cannot only read but also parse the file and generate the corresponding data structures.

The second thread consumes each bunch of lines and forks a set of worker threads in order to process them. The results are also queued in another shared list. Depending on the workload of the task to be performed and in order to maximize the overlapping time between I/O and computation routines, the worker threads can assume the responsibility of parsing the input text and creating the data structures. For example, the *effect* tool involves several network connections for querying CellBase, which are more expensive than text parsing; in this case, parsing is managed by the first thread. On the other hand, tests such as TDT or basic case-control association involve very simple computations. Therefore, the worker threads both create the data structures and compute the results. Another configurable argument that also allows to reduce the time that worker threads must wait is the number of lines in a bunch. Most of the application arguments related to parallelism can be tweaked in order to improve performance. At the moment, their value must be set via command-line or a configuration file, although possible ways of automation are being studied.

¹ Keepalived project website: <http://www.keepalived.org>

² HAProxy project website: <http://haproxy.1wt.eu>

³ Jersey library website: <http://jersey.java.net>

⁴ Hibernate project website: <http://www.hibernate.org/>

Fig. 2. HPG Variant *effect* sequence diagram

Finally, the third main thread consumes the list populated by the second one and writes the output to one file, as in statistical tests, or one per consequence type, as in the *effect* tool.

Additionally, a web application has been developed following the Software as a Service (SaaS) paradigm. Users can upload and permanently store their VCF data files in the machines we have deployed, conduct their studies, and visualize and download the results. It can be accessed at the URL <http://variant.bioinfo.cipf.es>. At the moment, only the *effect* tool is available this way, but *gwas* and HPG VCF Tools will also be accessible, providing the users with a full online solution. The web application will be described with more detail in

Sect. 2.2. The back-end can also be downloaded and run as a command-line standalone application, in case the user wants to avoid uploading huge files.

2.2 Interface

As stated at the beginning of Sect. 2, HPG Variant includes a collection of tools accessible via different user interfaces. This section describes the command-line interface in detail and how the web interface interacts with it.

The HPG Variant command-line is a combination of the name of the tool to invoke, a list of arguments available across the whole application, and another list of arguments specific to the tool.

Listing 1.1. Invoking hpg-variant

```
hpg-variant effect --vcf-file input1.vcf --region 1:10000-20000 --quality
40 --num-threads 4 --outdir path/to/output/folder
```

In Listing 1.1, the *effect* tool receives as input the file `input1.vcf`, which is filtered by a certain genomic region and a minimum quality. Four threads are used when parallelism is available and the results are written to the folder set in the `outdir` argument. When invoking the *gwas* tool, the test or analysis to conduct must be also specified as an argument, such as `--tdt` or `--fisher`.

All the tools in HPG Variant are able to filter their input. Some of the criteria available are region, quality and number of alleles, among others. These filters will be described in more detail in Sect. 4. In addition, other arguments used from the whole application are related to input and output, parallelization and memory management.

The application output varies for each tool. *Effect* generates a set of text files grouped by the consequence type of the variants, another one with all the consequence types found and a summary file with the count of each one. For a detailed explanation of the output see http://docs.bioinfo.cipf.es/projects/variant/wiki/Output_columns.

TDT analysis output is very similar to PLINK's. It shows the chromosome and position where the variant occurred, the possible alleles and the times they are transmitted, the odds ratio, the chi-square value and the p-value. Output of other tools has been suppressed for brevity.

Listing 1.2. GWAS TDT output

CHR	BP	A1	A2	T	U	OR	CHISQ	P-VALUE
1	742429	C	T	11	9	1.222222	0.200000	0.654721
1	767376	A	G	0	1	0.000000	1.000000	0.317311
1	769185	G	A	11	5	2.200000	2.250000	0.133614

Regarding the web application, it simplifies the use of the CLI by automatically generating and queuing the whole *effect* command. Moreover, it shows the

2. Summary

Filtered Variants [CHB.exon.2010_03.sites.vcf.filtered](#)

Genes with Variants [genes_with_variants.txt](#)

Consequence types histogram: [summary box](#)

3. Variants by Consequence Type [+info](#)

All (37953) [all_variants.txt](#)Consequence type description (5) [consequence_type_description.txt](#)

Intron variant (8765) [intron_variant.txt](#)

5KB downstream variant (2111) [5KB_downstream_variant.txt](#)

5KB upstream variant (2139) [5KB_upstream_variant.bed](#)

Exon variant (772) [exon_variant.txt](#)

[SNP \(4952\)](#) [SNP.txt](#)Regulatory region variant (14501) [regulatory region variant.txt](#)

LincRNA (254) [lincRNA.bt](#)

DNaseI hypersensitive site (1473) DNaseI hypersensitive site 1x1

Splice region variant (70) [splice_region_variant.txt](#)

CoG island (95) [CoG island.txt](#)

Coding sequence variant (192) [coding sequence variant](#)

Synonymous codon (R9) [synonymous_codon.txt](#)

RNA polymerase promoter (4EC) [RNA polymerase promoter](#)

Non-patented (198) [patented \(198\)](#)

Fig. 3. Some results of HPG Variant *effect* shown in its website

results in a more understandable way: (i) the summary section includes the analyzed VCF file (or a filtered version if applicable), a file with the genes that contain variants and a pie chart showing the ratio of occurrence of each consequence type, (ii) in another section users will find the consequence type files cited above and (iii) an embedded instance of the Genome Maps [12] browser allows to explore the genome along with the variants analysed (see Fig. 3). Also, results are kept in our servers and users can revisit or share them from any other computer.

3 HPG VCF Tools

HPG VCF Tools is a set of tools for preprocessing, filtering and manipulating of VCF files. The implementation aims to avoid excessive time consumption in tedious preprocessing tasks. Existing solutions are slow and consume a lot of memory due to their implementation in high-level languages like Perl.

At the moment, HPG VCF Tools includes parallel implementations for filtering, splitting and getting statistics. Merging will be included in the short-term.

The parallelization strategy of these tools is the same as the one described in Sect. 2.1.

The splitting tool (named *split*) allows to separate the original file into several ones, one per chromosome. Each of them is a valid VCF file. The statistics tool (*stats*) takes the data generated via the VCF library and performs some additional operations in order to enrich the data. As a result, the summary about a file contains the ratio between transitions and transversions or the average quality of the records in the file, among others. Finally, the filtering tool (*filter*) allows to chain several filters in order to get a set of variants that fulfill multiple conditions. Some filters take as input the results from the statistics tool. After filtering, two resulting VCF files are obtained, with the variants that passed and failed all filters, respectively.

4 VCF Handling Library

In order to set the foundations for HPG VCF Tools and HPG Variant, an efficient library for VCF file handling was implemented. Other existing applications tend to load whole files in main memory, making impossible to manage large files on systems without tens of gigabytes of RAM available per user.

VCF Library (*vcf-lib*) includes functionality for reading and writing VCF files, as well as the foundations for other tasks such as filtering and getting statistics. The file parsing has been implemented using the Ragel State Machine Compiler [13]. Similar parsers for GFF and PED files have been implemented using the same software.

Filters and statistics have been implemented as independent functions that can be easily embedded in data-parallel structures such as OpenMP for-loops. Available filters allow to remove variants not in certain regions, being (or not) SNP, having a quality or coverage lesser than specified or a number of alleles distinct than specified.

Statistics get information about each variant and a whole VCF file. The statistics of a variant show the times and relative frequency of appearance of the alleles and the genotypes than can be obtained when alleles are combined. The file statistics include the number of variants, samples, SNP, indels, bi-allelic and multi-allelic mutations, transitions and transversions in the file, and the number of variants that passed the filters applied.

5 Case Study

In this section we will illustrate a real case study using a dataset consisting of 49 trios, unaffected parents and an affected child, of sporadic short-segment Hirschsprung's diseased individuals. In total, 147 samples and 499,264 variants have been analyzed.

The analysis of sequencing data generally involves hundreds of thousands or even millions of variants. In order to avoid spurious associations in high error rate data, some quality control considerations must be noted before running the

analysis [14]. HPG VCF Tools *stats* describes variants in detail, making easier to detect low quality data. In this dataset we identified 1,092 missing variants in more than 20% of the samples and 36,428 variants with minor allele frequency (MAF) $<0.5\%$. These variants were discarded using HPG VCF Tools *filter*.

Since the analyzed individuals have been sampled from families, the most convenient of the analysis included in HPG Variant *gwas* is a family-based association study. The transmission disequilibrium test [15] showed a significant (p-value <0.05) association of Hirschsprung's disease with 23,012 variants.

Significant variants that result from the association test are rarely the direct causal mutations. Contrarily, the association can be direct, indirect or spurious. Direct or causal associations point to variants directly involved in the disease; indirect or non-causal associations appear when the variant is neighboring the causal locus but it is not directly involved, and spurious associations occur as a result of a poor quality control or an inadequate experimental design [16].

One of the strategies to identify causal variant is filtering the associated set by those that represent a higher risk in the integrity of the genome and its products. HPG Variant *effect* identifies variants located in functionally important places of the genome and the proteins (splice sites, transcription factor binding sites, non-synonymous changes, etc.). After running the analysis we detected a high number of variants with significant association placed at the RET proto-oncogene. The vast majority of these variants were located at the coding region and produced a change in the amino acid sequence, meaning that the final protein could be significantly different in Hirschsprung's diseased individuals. Interestingly, some authors have previously confirmed the association of RET as the major gene involved in this pathogenesis [17,18], which supports the validity of our approach.

6 Experimental Results and Comparisons

In order to guarantee the possibility of running this tools efficiently even in standard workstations, performance tests have been conducted on a Dell Precision T1600 system with an Intel Xeon Sandy Bridge E3 1245 processor and 8 GB on main memory. Depending on the tool analyzed, experimental results will be shown independently or in comparison with other applications.

A great part of HPG Variant *effect* execution time is spent on I/O operations, so it would be reasonable to think that the number of samples could have a great influence on performance. In Fig. 4 it is made obvious that this is not the case for typical datasets. Fig. 4 (a) shows the same linear time complexity with 3 and 147 samples, while Fig. 4 (b) shows the mean number of variants per second processed which is, in most of the cases, in the range between 1000 and 1200 variants.

The following benchmarks were executed using 3 main threads and another 2 worker threads.

When launching TDT against a file of 500,000 variants and 147 samples it was observed that PLINK and FBAT [19] spent about 1:30 minutes, whereas HPG Variant *gwas* ran for 20 seconds. UNPHASED [20] was unable to process the dataset in a reasonable amount of time.

Fig. 4. HPG Variant *effect*: time spent and mean variants per second

Association test exposed similar results. PLINK and GenABEL ran for about 1:20 minutes, whereas HPG Variant spent less than 25 seconds.

Fig. 5. Comparison of HPG Variant and HPG VCF Tools against other suites

Finally, the statistics tool from HPG VCF Tools spent about 20 seconds for the same file. PlinkSeq [21] and SNPTEST [7] spent 90 and 40 seconds, respectively, and VCFTools could not even process the dataset after 30 minutes. Since each sample must be checked in all variants, the number of samples has a great influence over performance. When this number decreases the tool speeds-up to the extent of spending just 18 seconds processing a file of 3.5M of reads and 3 samples. For this file, VCFtools could finish its execution after 8:47 minutes.

7 Conclusions and Future Work

Next Generation Sequencing technologies produce a vast amount of data that cannot be analyzed using current software, due to its sequential nature and careless memory management. A suite that includes essential features in genomic software has been implemented, with a good performance improvement both in time and memory consumption. In addition, a freely distributable library and a set of tools for Variant Call Format handling have also been developed, so the time spent in basic and repetitive tasks could also be reduced.

As future work, more analysis such as linear or logistic regression models for variant association are being implemented. Other areas like functional and pathway analysis will also be explored.

Regarding HPG VCF Tools, the collection of filters will be enriched with more interesting, from a biological point of view, filters. The splitting tool will accept a collection of criteria instead of just the chromosome of the variant. In addition, a VCF merging utility will be developed, taking care not just of bi-allelic variants but also multi-allelic ones, a feature not well-supported at the moment in the most widespread applications.

Finally, the VCF library will allow to parse compressed VCF files, improving storage saving. The possibility of parsing indexed VCF files is also being considered.

Acknowledgements. This work is supported by the grant BIO2011-27069 from the Spanish Ministry of Science and Innovation (MICINN) and PROMETEO/2010/001 from the Conselleria de Educaci3n of the Valencian Community. We also thank the support of the National Institute of Bioinformatics (www.inab.org) and the CIBER de Enfermedades Raras (CIBERER), both initiatives of the ISCIII, MICINN. CYG is supported by the Chair in Computational Genomics funded by Bull. This work has been carried out in the context of the HPC4Genomics initiative (<http://www.hpc4g.org>).

References

1. Majewski, J., Schwartzentruber, J., Lalonde, E., Montpetit, A., Jabado, N.: What can exome sequencing do for you? *J. Med. Genet.* 48(9), 580–589 (2011)
2. Danecek, P., Auton, A., Abecasis, G., Albers, C.A., Banks, E., DePristo, M.A., Handsaker, R., Lunter, G., Marth, G., Sherry, S.T., McVean, G., Durbin, R., 1000 Genomes Project Analysis Group: The Variant Call Format and VCFtools. *Bioinformatics* 27, 2156–2158 (2011)
3. Tarraga, J., Gonzalez, C.Y., Requena, V., Dopazo, J., Medina, I.: High Performance Genomics (HPG) Project, <http://docs.bioinfo.cipf.es/projects/hpg-project>
4. Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M.A., Bender, D., Maller, J., Sklar, P., de Bakker, P.I., Daly, M.J., Sham, P.C.: PLINK: a toolset for whole-genome association and population-based linkage analysis. *Am. J. Hum. Genet.* 81(3), 559–575 (2007)
5. Purcell, S.: PLINK v1.07, <http://pngu.mgh.harvard.edu/purcell/plink/>
6. Aulchenko, Y.S., Ripke, S., Isaacs, A., van Duijn, C.M.: GenABEL: an R library for genome-wide association analysis. *Bioinformatics* 23(10), 1294–1296 (2007)
7. Marchini, J., Howie, B.: Genotype imputation for genome-wide association studies. *Nat. Rev. Genet.* 11(7), 499–511 (2010)
8. Medina, I., De Mar3a, A., Bleda, M., Salavert, F., Alonso, R., Gonzalez, C.Y., Dopazo, J.: VARIANT: Command Line, Web service, and Web interface for fast and accurate functional characterization of variants found by Next Generation Sequencing. *Nucleic Acids Res., Web Server Issue* 40(W1), W54–W58 (2012), <http://docs.bioinfo.cipf.es/projects/variant/wiki/>

9. Bleda, M., Tarraga, J., De Maria, A., Salavert, F., Garcia-Alonso, L., Celma, M., Martin, A., Dopazo, J., Medina, I.: CellBase, a comprehensive collection of RESTful web services for retrieving relevant biological information from heterogeneous sources. *Nucleic Acids Res.*, Web Server Issue 40(W1), W609–W614 (2012)
10. Hindorff, L.A., Sethupathy, P., Junkins, H.A., Ramos, E.M., Mehta, J.P., Collins, F.S., Manolio, T.A.: Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *Proc. Natl. Acad. Sci. USA* (2009)
11. Bleda, M., Tarraga, J., De Maria, A., Salavert, F., Garcia-Alonso, L., Celma, M., Martin, A., Dopazo, J., Medina, I.: CellBase v1, <http://docs.bioinfo.cipf.es/projects/cellbase/wiki>
12. Medina, I., De Maria, A., Alonso, R., Salavert, F., Dopazo, J.: Genome Maps, a new generation of genome browser based on HTML5. Unpublished work, <http://genomemaps.org/>
13. Thurston, A.D.: Parsing Computer Languages with an Automaton Compiled from a Single Regular Expression. In: Ibarra, O.H., Yen, H.-C. (eds.) CIAA 2006. LNCS, vol. 4094, pp. 285–286. Springer, Heidelberg (2006)
14. Anderson, C.A., Pettersson, F.H., Clarke, G.M., Cardon, L.R., Morris, A.P., Zondervan, K.T.: Data quality control in genetic case-control association studies. *Nat. Protoc.* 5, 1564–1573 (2010)
15. Ott, J., Kamatani, Y., Lathrop, M.: Family-based designs for genome-wide association studies. *Nat. Rev. Genet.* 12, 465–474 (2011)
16. Clarke, G.M., Anderson, C.A., Pettersson, F.H., Cardon, L.R., Morris, A.P., Zondervan, K.T.: Basic statistical analysis in genetic case-control studies. *Nat. Protoc.* 6(2), 121–133 (2011)
17. Bolk Gabriel, S., Salomon, R., Pelet, A., Angrist, M., Amiel, J., Fornage, M., Attie-Bitach, T., Olson, J.M., Hofstra, R., Buys, C., Steffann, J., Munnich, A., Lyonnet, S., Chakravarti, A.: Segregation at three loci explains familial and population risk in Hirschsprung disease. *Nature Genet.* 31, 89–93 (2002)
18. Emison, E.S., McCallion, A.S., Kashuk, C.S., Bush, R.T., Grice, E., Lin, S., Portnoy, M.E., Cutler, D.J., Green, E.D., Chakravarti, A.: A common sex-dependent mutation in a RET enhancer underlies Hirschsprung disease risk. *Nature* 434, 857–863 (2005)
19. Laird, N., Horvath, S., Xu, X.: Implementing a unified approach to family based tests of association. *Genet. Epidemiol.* 19(suppl. 1), S36–S42 (2000)
20. Dudbridge, F.: Likelihood-based association analysis for nuclear families and unrelated subjects with missing genotype data. *Hum. Hered.* 66, 87–98 (2008)
21. Purcell, S.: PLINK/SEQ v0.08, <http://atgu.mgh.harvard.edu/plinkseq/>